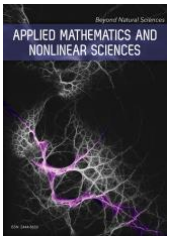
sciendo Applied Mathematics and Nonlinear Sciences, 9(1) (2024) 1-16



Applied Mathematics and Nonlinear Sciences

https://www.sciendo.com

Intelligent big data visual analytics based on deep learning

Ruixiang Guo[1,†]

1. Information Construction and Management Office, Minnan Normal University, Zhangzhou, Fujian, 363000, China.

Submission Info

Communicated by Z. Sabir
Received March 26, 2023
Accepted June 28, 2023
Available online December 18, 2023

Abstract

In this paper, we first constructed a deep learning model, optimized the LSTM model to get the BiLSTM model based on the long and short-term memory network, and used the generative adversarial network to calculate the probability distribution of data. Then, the advantages of deep learning in intelligent big data visualization and analysis are explored from the dimensions of data preprocessing, dimension anchor layout, coordinate expansion and data analysis. Finally, the efficiency of the deep learning model is compared with that of other algorithms using indicators such as accuracy and recall, and the feasibility of this paper's method is verified by empirical analysis using intelligent transportation data as an example. The results show that the model in this paper achieves an accuracy rate of 95.5%, the loss rate is stable at 0.2% to 0.4%, and the average running time is maintained at 20ms, which are all better than other models. The predicted and real values of traffic data for the Deep-STCL model using deep learning basically match, indicating that the deep learning model has obvious advantages in data visualization and analysis.

Keywords: Deep learning; LSTM model; Generative adversarial network; Long short-term memory network; Data visualization.
AMS 2010 codes: 68P01

†Corresponding author.
Email address: 13960073548@163.com

ISSN 2444-8656
https://doi.org/10.2478/amns.2023.2.01539

1 Introduction

In the era of big data, people's attention and application of data have reached an unprecedented degree, and the data generated by the Internet and other emerging fields exceeds the scale, structure, and difficulty of analyzing traditional data. And in this context, a new science-data science has emerged [1]. Data visualization technology involves the use of computer graphics and image processing technologies, in which data is replaced with graphics or images for display and interactive processing. Data visualization covers a variety of aspects, such as interactive technology, computer vision, graphics, etc. It can intuitively display the basic features and hidden information of high-dimensional data, so for the analysis of high-dimensional data, its visualization is undoubtedly the most convenient and fast method, and the data will be displayed graphically to be closer to the user's habits and use of the needs of the user [2-3]. Data visualization technology is an effective means of data understanding and analysis, for the analysis of high-dimensional data is of great help. It can be found from the high-dimensional data collection of data implied laws and obtain useful information to help users understand and analyze the data so as to provide the basis for users to make the correct decisions and judgments [4].

Tang, Y. et al. used visual analysis techniques to analyze traffic data, traffic flow data to guide business operations, and thematic models to analyze taxi trajectory data to determine foot flow [5]. Wang, C. et al. studied a visualization tool based on text markers and used the tool to analyze the massive text data generated by the software system [6]. Lundqvist, K. et al. studied the visual analysis technology based on GIS technology, used this technology to analyze three-dimensional archaeological virtual reality spatial data, and analyzed the feasibility of this technology through actual cases [7]. Chen, M. et al. use data visualization techniques to analyze the causality of data to help people understand the causality of data and make better decisions [8]. Zhongheng, Zhang et al. used visual data analysis technology to predict and analyze medical data to provide personalized medicine services and tap new development potential in the medical field [9]. Huang, J. et al. used high-performance monitoring tools to improve and support the resource allocation ability of the data analysis system in view of the characteristics of data analysis, which effectively improved the performance of the data visualization and analysis system [10].

Tobias, P. et al. used image analysis algorithms to visualize medical information, and proposed ImgLib2, a Java library for n-dimensional data representation and manipulation, to optimize the performance of data visualization analysis [11]. Song, J. et al. used classification analysis and cluster analysis to analyze construction engineering data, and established a digital analysis technology for construction sites, which brought a digital revolution to construction sites [12]. Hurtut, T. et al. studied a visual analysis technique for analyzing high-frequency trading, using this technology to analyze the relationship between portfolio options and returns and risks, and promote the development of high-frequency trading and risk control [13]. Machado, J. A. T. et al. conducted a visual analysis of seismic data based on information technology and used cluster analysis to generate visual maps to visually and usefully represent seismic data, effectively analyze the degree of earthquake damage and effectively predict earthquakes [14]. Quince, C. et al. established a new data visualization analysis platform, which can quickly extract multi-dimensional information of each configuration, providing a convenient and reliable path for data exploration [15].

In this paper, we first establish a general framework for deep learning models, use long and short-term memory networks to portray the relationship between sequence output and previous information, establish a generative adversarial network for conditional probability distributions to determine the category to which the variables belong, and use the model's attentional mechanism to augment some of the weights in the input data of the neural network model. Then, the deep learning model is used to carry out data visualization analysis, preprocess the data based on the principles of min-max

normalization and Z-score normalization, project the principal components of the dataset on the Radviz circumference, and carry out anchor point layout, axis expansion and data visualization analysis. Finally, in order to verify the feasibility and effectiveness of the model in this paper on data visualization, the data analysis performance of the deep learning model is compared with SVM, clustering and other models, comparing the basic performances of the different models such as accuracy, recall and F1 value, and analyzing the different performances of the different branch models of deep learning on traffic data prediction and visualization analysis.

2 Deep Learning Models

2.1 Long- and short-term memory networks

Recurrent neural networks can portray the relationship between the output of the current moment of a sequence and the previous information, with memory characteristics, and have applications in speech recognition, machine translation, and temporal analysis and prediction.

Recurrent neural networks can make full use of information from previous moments, but they also pose a technical challenge in that they are characterized by long-term dependencies [16]. The long-term dependency problem can lead to recurrent neural networks not being able to learn dependencies that span a wide range. With improved research by scholars, new models have emerged, among which long and short-term memory networks are the most commonly used. The LSTM model adds a state space and three gating units to the structure of the recurrent neural network, f_t , i_t , and o_t represent the forgetting gate, input gate, and output gate, respectively, and f_t and i_t are the keys for the model to memorize the long-term information.

In the network structure, the result of the forgetting gate requires a joint decision of x_t , h_{t-1} and c_{t-1} , which is used to determine the information to be discarded. And the role of the input gate is to decide which part of the information will enter the current moment state and get the final output result of the neural unit. The three gating units and the specific representation of the state space are as follows:

$$f_t = \sigma\left(w_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(w_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$\tilde{c}_t = \tanh\left(w_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{3}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{4}$$

$$O_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t \cdot \tanh\left(c_t\right) \tag{6}$$

The above is the structure of the long short-term memory network and the calculation formula of the three gating units, which ensures the long-interval dependence of the long-term, short-term memory

network through freely selected memory and abandoned information and has strong processing ability in time series problems.

On the basis of the LSTM model, the BiLSTM model is improved, which includes two parts: forward LSTM and reverse LSTM model, in which the forward LSTM model is used to process the original sequential input and mine the above data information, and the reverse LSTM model processes the reverse order input information, mining the data information contained below, and can better capture the context information compared with the long short-term memory network model.

2.2 Generating Adversarial Networks

Machine learning contains two main categories of models: generative and discriminative. Among them, discriminative models are models built for conditional probability distributions to determine the class to which a variable belongs, while generative models are models built for joint probability distributions.

The value function of the generative adversarial network is:

$$\min_{G}\max_{D}V(D,G)=E_{x\sim P_{data}}[\log D(x)]+E_{z\sim P_z}[\log(1-D(G(z)))] \tag{7}$$

In equation (7), E represents the expectation value, p_{data} is the distribution of the real data, x comes from P_{data}, and the output is $D(x)$ after the discriminator D. P_Z is the distribution of the generated data, Z comes from P_Z after the generator, and the output is $G(z)$.

In the choice of generator model and discriminator model, according to the data characteristics and model effect, this paper chooses GRU as the generator and convolutional neural network as the discriminator in the implementation of the generative adversarial network [17]. Next, the models of the generator and discriminator are mainly introduced.

Cho et al. proposed an improved model-gated recurrent neural network based on recurrent neural networks. The main difference between it and the LSTM model is the number of gated units; the rest of the difference between the long and short-term memory network is very little, and it is also known as a variant of the LSTM model. The GRU model and the LSTM model have the same overall concept, and the specific expressions are as follows:

$$Z_t=\sigma\left(w_z\cdot\left[h_{t-1},x_t\right]+b_z\right) \tag{8}$$

$$r_t=\sigma\left(w_r\cdot\left[h_{t-1},x_t\right]+b_r\right) \tag{9}$$

$$h_t=\tanh\left(w_h\cdot\left[h_{t-1},x_t\right]+b_h\right) \tag{10}$$

$$h_t=\left(1-Z_t\right)\cdot h_{t-1}+Z_t\cdot h_t \tag{11}$$

In the expressions (8)-(11), Z_t is the update gate, r_t is the reset gate, h_t is the memorized content of the current moment, and h_t is the final memorized result of the current moment.

A convolutional neural network consists of five main layers. The input layer is the beginning of the whole network. When dealing with stock data, it is necessary to process the time series data into a type that meets the GRU model acceptance. The data processing consists of wrapping the data into a three dimensional matrix containing the number of length widths. The convolutional layer is a key component that further extracts information from the neural network through the convolutional kernel to find features of the stock data with a higher level of abstraction. The pooling layer is able to compress the size of the matrix output from the previous layer to improve the speed of the model and, at the same time, can have the effect of preventing overfitting. The convolutional and pooling layers extract the high-level features from the stock data, and then the two layers, the softmax layer and the fully connected layer, are mainly responsible for the classification of the features after extraction. The softmax layer is designed to convert the output of a neural network into a probability distribution and calculate the probability of belonging to different categories to complete the classification.

2.3 Attention mechanisms

Attentional mechanisms are techniques that mimic human cognitive attention and can enhance the weight of some parts of the input data to a neural network model while diminishing the weight of other parts as a way of focusing the network's attention on a small portion of the data that is most important.

Assuming that there is a sequence of tokens arranged at index i, for each token i, the neural network computes a corresponding non-negative soft weight w_i that satisfies $\sum_i w_i = 1$. Each token corresponds to a vector v_i obtained from word embeddings, and the weighted average $\sum_i w_i v_i$ is the output of the attention mechanism.

Soft weights can be computed using the query-key mechanism. From the word embedding of each token, the corresponding query vector q_i and key vector k_i are computed, and the corresponding weights are obtained by computing the softmax function of the dot product $q_i k_i$, where i represents the current token and j represents the tokens that have an attentional relationship with the current token.

Denote N sets of input information by $X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N}$, where the D-dimensional vector $x_n \in \mathbb{R}^D, n \in [1, N]$, denotes a set of input information. In order to avoid the waste of computational resources, only the critical information from X is selected as input. The attention mechanism first calculates the attention distribution of the input information and then performs a weighted average.

In order to select the information that is closely related to the task from the input information $[x_1, x_N]$, task-related query vectors are defined and a scoring mechanism is used to query the relevance of the vectors to the input vectors.

Given a query vector q associated with a particular task, denote the index position of the selected information by the attention variable $z \in [1, n]$, i.e., $z = n$ indicates that the nth input vector is selected. Calculate the probability α_n of selecting the ith input vector given q and X:

$$\alpha_n = p(z = n \mid X, q) = \text{softmax}\left(s\left(x_n, q\right)\right) = \frac{\exp\left(s\left(x_n, q\right)\right)}{\sum_{j=1}^{N} \exp\left(s\left(x_j, q\right)\right)} \tag{12}$$

Where α_n is called the attention distribution and $s(x, q)$ is the attention scoring function, which can be computed in several ways.

1) Additive modeling:

$$s(x, q) = v^{\cdot} \tanh(Wx + Uq) \tag{13}$$

2) Dot product modeling:

$$s(x, q) = x^{\cdot} q \tag{14}$$

3) Scaled dot product model:

$$s(x, q) = \frac{x^{\cdot} q}{\sqrt{D}} \tag{15}$$

4) Bilinear modeling:

$$s(x, q) = x^{\cdot} Wq \tag{16}$$

Where W, U, v is the model parameter to be learned and D is the dimension of the input vector.

Assuming $W = U^{\cdot} V$, the formulation of the bilinear model can be written as:

$$s(x, q) = x^{\cdot} U^{\cdot} Vq = (Ux)^{\cdot} (Vq) \tag{17}$$

That is, the dot product is computed after linear transformation of x and q respectively. Compared to the dot product model, the bilinear model introduces asymmetry in the calculation of similarity.

3 Visual analysis of data based on deep learning

3.1 Data pre-processing

When visualizing and analyzing data, certain data preprocessing is required, such as the two algorithms proposed in this paper, both of which require data standardization. Data standardization is to transform the data in the same way, so that it is in a specific interval, unified data level, so as to facilitate weighting and comparison.

The standardization methods chosen in this paper are min-max standardization and Z-score standardization.

1) min-max standardization

This standardization method will be a linear transformation of the data; the data will fall within the [0,1], so it is also called 0-1 standardization. This standardization method will not have an impact on the original distribution of data. The distribution of data before and after the normalization is consistent, the specific transformation formula is shown in equation (18):

$$x = \frac{x - \min}{\max - \min} \tag{18}$$

Where $\max$ is the maximum value of this data set and $\min$ is the minimum value, the calculation is relatively simple. It is also easier to reduce the standardized data, but if the data set is modified, there will be a change in the maximum and minimum values, which needs to be recalculated. Since the C-Radviz method proposed in this paper needs to rely on the original distribution information within the data and needs to retain the data distribution to the maximum extent, this standardization method is used.

2) Z-score normalization

This standardization method is essentially mean-variance normalization and is the most common standardization method. After standardization, the dataset will conform to a normal distribution with a mean of 0 and a standard deviation of 1. This standardization method is often used in clustering algorithms that derive similarity by distance, and since PCA-Radviz does not rely on information about the internal distribution of the data, it is therefore adopted as a standardization scheme, and the specific transformation the formula is shown in equation (19):

$$x^* = \frac{x - \mu}{\sigma} \tag{19}$$

Where μ is the mean size of the dataset and σ is the standard deviation size of the dataset.

In the PCA-Radviz method, the data set was processed using Z-score normalization and the processed data was deposited into the matrix M as follows.

Calculate the mean μ and standard deviation σ of the original data:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{20}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{21}$$

The transformation function after standardized treatment was calculated according to μ and σ as in equation (21), and the processed data conformed to the standard normal distribution, i.e., the mean was 0, the standard deviation was 1, and the range of variation of the data after standardized treatment was $-1 \le x^* \le 1$.

3.2 Dimension Anchor Layout and Data Visualization

When dimension sorting and uneven distribution are performed, the data dimensions originally distributed uniformly on the circumference are turned into the location of the principal component

projection, and since the dataset has a better characterization effect in the direction of the principal component, it will theoretically have a better visualization effect when it is projected on the Radviz circumference as well, and the detailed anchor point layout method and axis expansion steps are as follows.

1) is analyzed to visualize the projection plane of Radviz consisting of the two-dimensional vectors with the largest contribution to the x and y axes.

Calculate the covariance matrix $Cov\left(X_i, Y_j\right)$ of the normalized matrix M to obtain the eigenvalues $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ and eigenvector $C = \{c_1, c_2, \cdots, c_k\}$ of the covariance matrix, and then sort the eigenvectors from largest to smallest. The first vector selects the direction with the largest variance calculated by the original data, the second vector selects the direction in the plane orthogonal to the first vector that makes the variance the largest, the third vector is the direction with the largest variance in the plane orthogonal to the first and second vectors, and so on, to obtain the ranking of the k eigenvectors, and then the eigenvalues λ are arranged in order from largest to smallest, so as to give the importance level of the components. Select the eigenvectors corresponding to the two eigenvalues with the largest eigenvalues as the x and y axes to form the Radviz visualization projection plane L.

2) Find the cosine distance of the original dimension of the data from the x and y axes to get the projection angle and vector length of the dimension anchor point.

Fig. 1 shows the projection of the original dimension on the two-dimensional plane, it can be seen that in the plane L, respectively, the cosine distance $\cos\theta_{i1} = \frac{\vec{a_i} \cdot \vec{x}}{\left\|\vec{a_i}\right\| \cdot \|\vec{x}\|}, \cos\theta_{i2} = \frac{\vec{a_i} \cdot \vec{y}}{\left\|\vec{a_i}\right\| \cdot \|\vec{y}\|}$ of each dimension of the original data and the x and y axes of the projection plane L are obtained to get the angle and vector length of the original dimension projected on this plane L, that is, it constitutes the coordinates of the dimensional anchors of the original dimension on the projection plane L $\left(x_i, y_i\right)$.

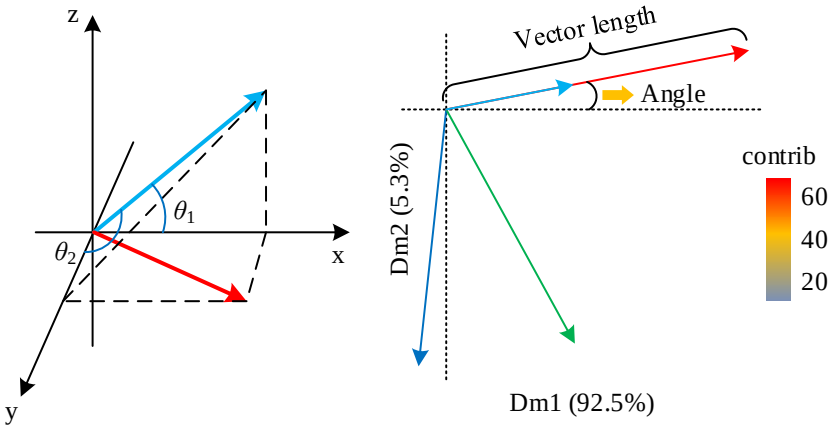


Figure 1. Projection of original dimension on two-dimensional plane

3) Designing a reasonable layout of dimension anchor points for the Radviz circle using the projection angle.

In order to make the coordinates of the dimension anchor points fall on the Radviz unit circle, the length of the obtained dimension anchor point vectors is telescoped and transformed to the unit length, and the angle remains unchanged, and the position of the Radviz dimension anchor points on the Radviz circumference is calculated as in equation (22):

$$\left(\frac{x_i}{\sqrt{x_i^2 + y_i^2}}, \frac{y_i}{\sqrt{x_i^2 + y_i^2}} \right) \tag{22}$$

4) Project the raw data points inside the Radviz unit circle in a dot pattern.

The projection in this step is consistent with the original Radviz method. The data is projected inside the circle as small dots, and the size of the data on the dimension is proportional to the size of the spring pull of the anchor point of the dimension on the dot, the dot finds the equilibrium position under the spring pull of multiple anchors, and the coloring is done according to the category of the data in the dataset in order to enhance the visualization.

5) Extend the z-axis to pull the 2D planar point into the 3D stereo

According to the Radviz 2D plane visualization effect in step (d), calculate the Euclidean distance from the original data point to the origin of the circle, and use it as the value of the z-axis to lift the 2D plane point into the 3D stereo to further enhance the display dimension of the data:

$$\frac{x_1 - a_1}{u_1} = \frac{x_2 - a_2}{u_2} = \cdots = \frac{x_n - a_n}{u_n} \tag{23}$$

$$x_{i+1} = l_i x_i + b_i \tag{24}$$

3.3 Parallel Coordinate Visualization

The parallel coordinate visualization technique is a typical geometry-based multidimensional visualization technique that represents n-dimensional data in a two-dimensional form. It is widely used in the field of data mining and visualization. Its basic idea is to map n attributes of n-dimensional data into two-dimensional space by means of n equidistant parallel axes, in which each parallel axis represents the dimension of high-dimensional data, and the values of these axes are the size of the attribute values, and the dimensions are connected by means of folding lines so that the user can see the distribution of the high-dimensional data directly in two-dimensional space through this visualization method.

Assuming that the dimension of the high-dimensional data is n-dimensional, then there are n parallel y-axes in the parallel coordinate visualization, the spacing between them is equal, and the points of the high-dimensional data will be displayed as countless intersecting polylines on the parallel coordinate axes in the two-dimensional space. There is a one-to-one correspondence between the points of the high-dimensional data and the polylines. For n-dimensional data, a point mapped into a fold line satisfies the following equations (25) and (26):

$$\frac{x_1 - a_1}{u_1} = \frac{x_2 - a_2}{u_2} = \cdots = \frac{x_n - a_n}{u_n} \tag{25}$$

Its deformation is followed by equation (26):

$$x_{i+1} = l_i x_i + b_i \tag{26}$$

4 Model data visualization performance analysis

4.1 Basic model performance analysis

4.1.1 Model Accuracy, Recall and F1 Value Analysis

In this paper, four models, SVM, RankSVM, K-mean clustering, and plain Bayes, are selected for comparison and evaluation with the models. For the evaluation indexes of the models, this paper uses four evaluation indexes for evaluation, which are Accuracy Acc, Precision Pre, Recall Rec, and F1 score.

Table 1 shows the accuracy results of this paper and the compared models under the test set. From the results shown in Table 1, it can be seen that under the four evaluation metrics of Acc, Rec, F1, and Pre, the model of this paper can achieve an overall accuracy of 95.5%, and the other metrics also achieve at least 93.3%, which are better than the other models.

Table 1. Model evaluation results

Model	Acc	Rec	F1	Pre
SVM	0.852	0.849	0.836	0.845
RankSVM	0.871	0.845	0.845	0.832
K mean clustering	0.887	0.891	0.855	0.872
Plain beth	0.915	0.906	0.914	0.904
This article	0.955	0.943	0.937	0.933

In order to better compare the advantages and disadvantages of different models, this paper plots the accuracy and loss rate graphs of the validation set. Figure 2 shows the comparison of the trend of the accuracy rate of the validation set, and Figure 3 shows the trend of the loss rate of the validation set.

From Fig. 2, it can be seen that the accuracy rate of each model shows a rising to smooth change. Under the first 4 epochs, the models fluctuate a lot, and in the smooth phase, all models reach more than 80% accuracy, and two models reach more than 90%. However, the models in this paper can reach accuracy above 90% and maintain stability under fewer epochs, which indicates that the models in this paper have better classification ability.

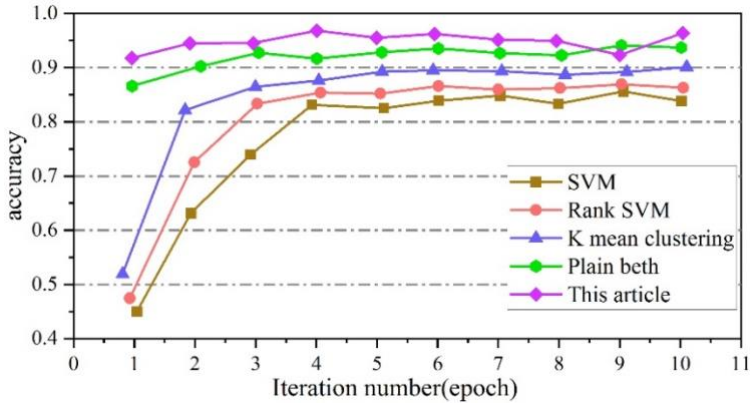


Figure 2. The comparison of the trend of accuracy changes

The loss rate of the model in this paper is minimized and stabilized in the range of 0.2% to 0.4% compared with other comparative models, which indicates that the model in this paper has better convergence. The loss rate only fluctuates once, which may be the effect of a sudden change of some unknown factors, and it does not affect the overall trend of smooth change. Overall, the model in this paper is better than the comparison model in terms of convergence speed and accuracy.

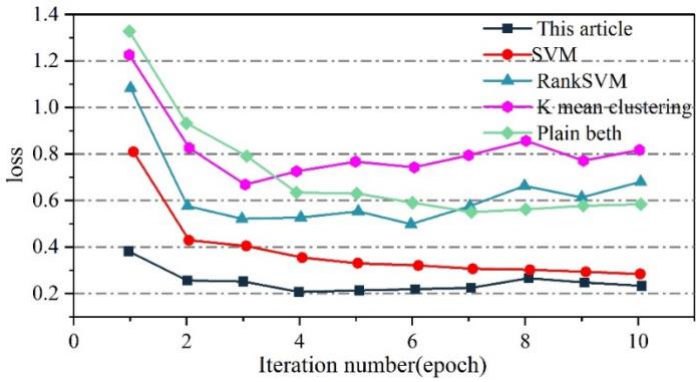


Figure 3. The trend of loss rate

4.1.2 Comparative analysis of average running time

Figure 4 shows the comparison of the algorithm runtime for each model with different sampling ratios in the Amazon dataset. It is clear that the SVM, RankSVM and deep learning algorithms running time hardly increases with the increase of sampling proportion, basically maintains around 50ms, and their average time overhead is much smaller than the K-mean clustering, plain Bayes algorithms running time increases with the increase of sampling size.

The results show that K-mean clustering, plain Bayesian algorithm method is simple, but the sampling performance is the worst, as the analysis is carried out K-mean clustering running time is even between 100ms-500ms, in the sampling, the nodes themselves are not taken into account in the slightest degree of variability. SVM, RankSVM, algorithms as a result of the use of the proximity distribution based on the hierarchical strategy, so that the algorithms have a small sampling error, sampling performance is good. Using deep learning algorithms to obtain the approximate degree distribution of nodes, the asymptotic time complexity of the deep learning algorithm is linear time, which ensures that the algorithm can be completed in linear time, and the deep learning algorithm running time does not increase with the increase in the sampling ratio, the average running time basically maintains between 20ms, the lowest running time and the best performance.

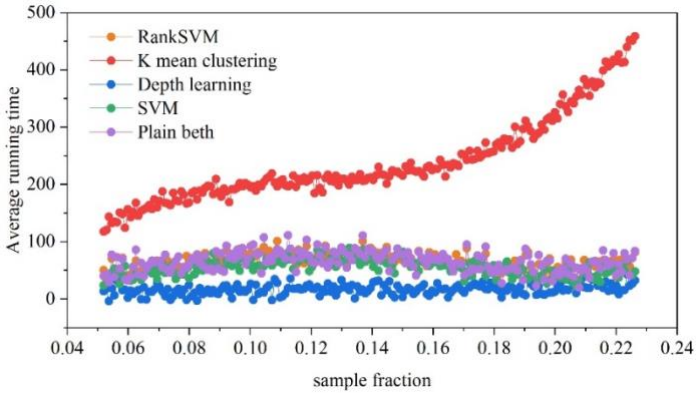


Figure 4. Average running time contrast

4.2 Empirical analysis

4.2.1 Analysis of results of predictive evaluation indicators

In the empirical analysis, this paper takes the intelligent traffic data of City A as the research object and uses the deep learning model to make travel prediction and visualization analysis of the traffic data. Since the deep learning short-time travel demand prediction model has three branches, each branch can get the prediction results of the corresponding branch. Therefore, in addition to the LSTM-based prediction framework and the CNN-based prediction framework, there are three other baseline models in the experiment. They each utilize a temporal property in the travel demand data, but the network structure is a branch of the whole framework. In the following analysis of the experimental results, these three baseline models are named CLC, CLP, and CLT. CLC stands for the Closeness branch of deep learning, CLP stands for the Period branch of deep learning, and CLT stands for the Trend branch of deep learning.

The experiments verify the performance of the six deep-learning branching models under the three evaluation metrics of RMSE, MAE, and MAPE. Table 2 shows the comparison of the RMSE results of the six models. Table 3 shows the comparison of MAE results for the six models. Table 4 shows the comparison of MAPE results for the six models.

It can be seen that the performance of the models is different at different points in time, which is due to the different densities of travel demand in the city at different points in time. For example, since 6 a.m. is too early in the morning, the density of travel demand is not very high, so the travel demand in many geographic blocks is 0. Therefore, the value of the evaluation metrics decreases as the cumulative error of the model prediction decreases. Conversely, as the travel demand increases, the model predicts more samples, the cumulative error increases, and the value of such metrics increases. Meanwhile, comparing the values of the three evaluation metrics at different times of the day separately, it can also be seen that the values of the Deep-STCL model are smaller than those of the other baseline models, which also proves that the full utilization of spatio-temporal characteristics of the model brings about an improvement in accuracy. In addition, from the perspective of branching results and synthetic results, it can be observed that Deep-STCL outperforms the three branching models, namely CLC, CLP, and CLT. This result demonstrates the effectiveness of model fusion in the prediction process.

Table 2. The comparison of rmse results of the six models

Time	6:00	9:00	12:00	15:00	17:00	20:00	23:00
LSTM	1.718	4.224	3.539	4.001	3.927	4.214	3.056

CNN	1.434	3.640	3.258	3.581	3.358	3.761	3.727
CLC	1.225	3.226	3.161	3.688	3.610	3.823	3.629
CLP	1.175	4.576	3.237	3.401	3.633	3.302	2.726
CLT	1.285	3.371	3.052	3.396	3.342	3.054	3.104
Deep-STCL	1.222	3.151	2.973	3.234	3.113	2.912	2.709

Table 3. Comparison of model mae results

Time	6:00	9:00	12:00	15:00	17:00	20:00	23:00
LSTM	0.874	1.498	1.159	1.238	1.378	1.218	0.805
CNN	0.565	1.580	1.197	1.239	1.289	1.227	1.038
CLC	0.374	1.123	1.104	1.320	1.258	1.266	1.005
CLP	0.345	1.450	1.167	1.144	1.182	1.162	0.817
CLT	0.379	1.364	1.097	1.191	1.178	1.053	0.867
Deep-STCL	0.355	1.033	1.037	1.060	1.130	0.996	0.725

Table 4. Six model mape results contrast

Time	6:00	9:00	12:00	15:00	17:00	20:00	23:00
LSTM	18.22	33.43	32.59	32.08	31.01	28.06	18.22
CNN	13.86	24.11	22.79	22.74	21.75	18.88	13.86
CLC	11.55	18.55	22.64	20.68	22.38	20.18	11.55
CLP	12.42	20.24	23.27	20.30	20.88	18.89	12.42
CLT	11.86	18.84	22.11	20.11	21.16	19.08	11.86
Deep-STCL	11.22	18.46	21.13	19.35	20.21	17.89	11.22

4.2.2 Comparative analysis of visualization of model prediction results

In order to observe the prediction of each model more intuitively, this subsection predicts the travel demand in city A for 24 hours a day and then selects a geographic block from it and plots the comparison between the prediction results of different models and the real travel demand. The comparison curve of real travel demand with predicted demand of different models is shown in Figure 5.

The horizontal coordinates in the graph represent different parts of the day, while the vertical coordinates represent the size of the travel demand. It can be seen that the Deep-STCL model not only has a higher prediction accuracy, but also the prediction result of the model is the same as the trend of the real travel demand. The LSTM model has a higher prediction accuracy in the early stage, but after 12 noon, the prediction accuracy of the model fluctuates a lot, and the model can't depict the trend of the order demand change in the second half of the day well. The prediction accuracy of the LSTM model is the highest, with deviations between 20 and 100 from the correct value. The CNN model also has a high prediction accuracy in the early stage, but in terms of the overall prediction, the model's prediction has a large deviation from the correct value with a large deviation of 20-50, and it cannot accurately depict the change in demand for order travel in the whole day. Overall, the Deep-STCL effect in deep learning has a better prediction ability than the two baseline models, with an error of less than 5 from the correct value, almost matching the correct value. The Deep-STCL model has good prediction accuracy, and the trend of travel demand changes at different moments of the day.

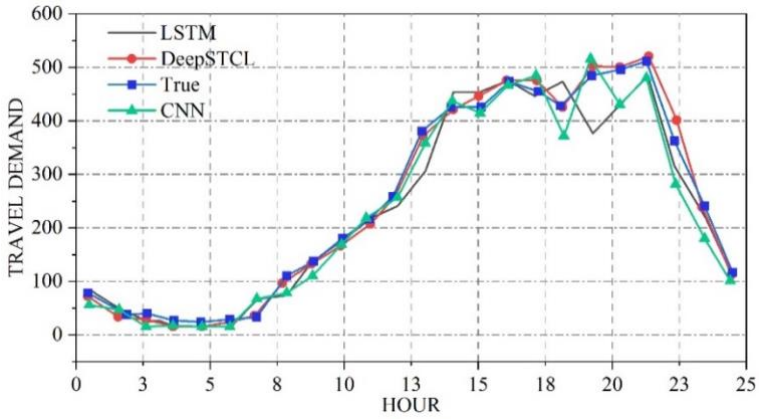


Figure 5. Real travel requirements and different models predict demand contrast diagram

Next, the top 15 blocks with the highest number of travel demands are extracted from all the geographic blocks, and 12:00 noon peak is selected as the predicted moment from one day, and it is predicted using the Deep-STCL model. Then, the predicted and real values are compared and analyzed, and Figure 6 shows the comparison between the real and predicted values of travel demand at 12:00 noon for the geographic blocks. The horizontal coordinate in Figure 6 represents the coding of the geographic block, while the vertical coordinate represents the value of the specific travel demand in the block. Each geographic block corresponds to two metrics: the predicted value and the actual value. The Deep-STCL model performs very well in the top 15 blocks with the highest travel demand, and there is not much difference between its predicted value and the true value, basically only about 5-10.

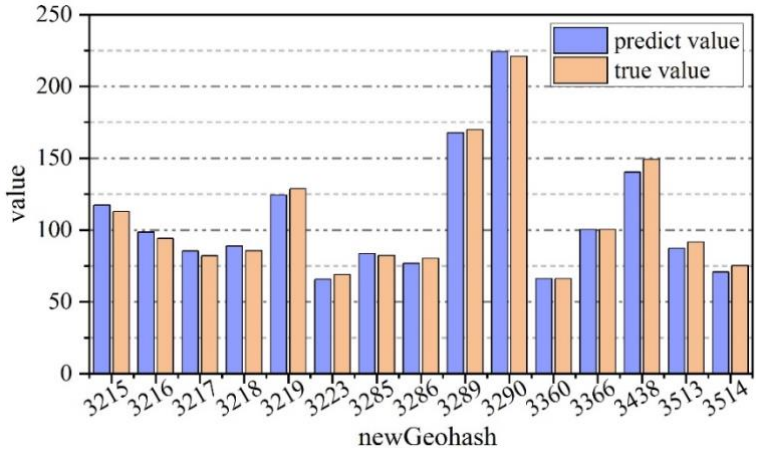


Figure 6. The actual value of the travel demand at 12:00 noon is compared to the predicted value

Meanwhile, in order to verify whether the overfitting phenomenon occurs in deep learning, this paper plots the changes in the loss function during the training and testing phases. Figure 7 shows the loss function during the training and testing phases.

It can be seen that in the training phase and the testing phase, the trend of the loss function decline curve is very similar; there is no sudden jump, and the training error of this paper's method is basically no more than 5% between the training error and the training value error. Meanwhile, since the sample number of the test dataset is smaller than that of the training dataset, the loss function curve of the test set is located below the training set. Overall, the loss variations in the training and test phases are similar, and there is no overfitting. In conclusion, the Deep-STCL model in deep learning achieves

good results in both traffic data prediction and visualization analysis, indicating that visualization techniques based on deep learning models are feasible and effective.

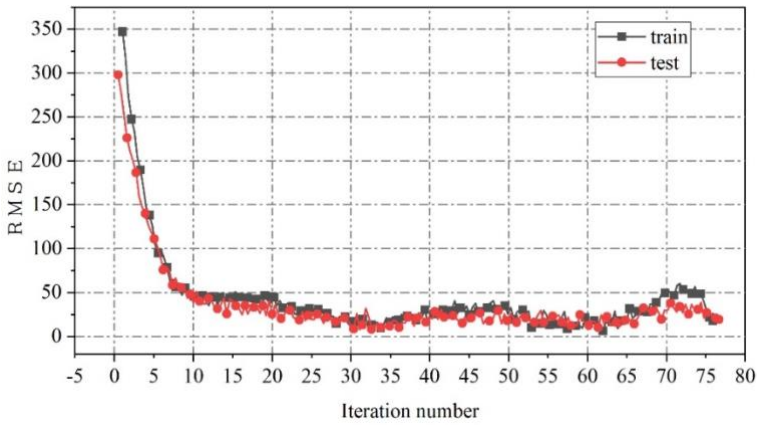


Figure 7. Training stage and test stage loss function

5 Conclusion

This paper is based on deep learning models for data visualization analysis, analyzes the basic performance of each model for visualization analysis by comparing it with other models, and studies the performance of different branches of deep learning models on data visualization by taking the traffic data of A urban area as an example. The primary results are displayed as follows.

1) The accuracy, recall and F1 values of SVM, RankSVM and K-mean clustering models are basically distributed between 0.83-0.88, and the Acc, Rec, F1 and Pre index values of the plain Bayesian model are distributed between 0.9-0.91, and the results of evaluation indexes of the deep model are above 0.93, which indicates that the model of this paper has the best performance on data visualization.

2) In terms of running time, the running time of other models is basically between 50-500ms, while the running time of the deep learning model is between 30ms. In terms of model loss rate, the model in this paper has the smallest loss rate and stabilizes in the interval of 0.2% to 0.4% compared with other comparative models and has better convergence.

3) In the visualization and analysis of intelligent traffic data in City A, the deep learning branch models all achieve good results, and the Deep-STCL model outperforms the other deep learning branch models in the model prediction visualization and analysis and the evaluation indexes of RMSE, MAE, and MAPE.

Funding:

This research was supported by a research project on educational informatization in Fujian Province's universities (FJGX22005).

References

[1] Nonaka, T. (2011). Visual data analysis. (visualization software). Acta Polytechnica Hungarica, 8(2), págs. 215-217.

[2] Mcnamee, R. L., & Eddy, W. F. (2001). Visual analysis of variance: a tool for quantitative assessment of fmri data processing and analysis. Magnetic Resonance in Medicine.

[3] Corinna, Vehlow, David, P., Kao, Michael, R., & Bristow, et al. (2015). Visual analysis of biological data-knowledge networks. BMC bioinformatics

[4] Michael, Golden, Darren, & Martin. (2012). Dooss: a tool for visual analysis of data overlaid on secondary structures. Bioinformatics.

[5] Tang, Y., Sheng, F., Zhang, H., Shi, C., Qin, X., & Fan, J. (2018). Visual analysis of traffic data based on topic modeling (chinavis 2017). Journal of Visualization, 21(4), 661-680.

[6] Wang, C., Jiang, J., Daneva, M., & Van Sinderen, M. (2023). Coolted: a tool for co-labeling and visual analysis of textual dataset. Science of Computer Programming.

[7] Landeschi, G., Dell'Unto, N., Lundqvist, K., Ferdani, D., Campanaro, D. M., & Touati, A. M. L. (2016). 3d-gis as a platform for visual analysis: investigating a pompeian house. Journal of Archaeological Science(65-).

[8] Chen, M., Trefethen, A., Banares-Alcantara, R., Jirotka, M., Coecke, B., & Ertl, T., et al. (2011). From data analysis and visualization to causality discovery. Computer, 44(10), 84-87.

[9] Sven, Poucke, V., Zhongheng, Zhang, Martin, & Schmitz, et al. (2016). Scalable predictive analysis in critically ill patients using a visual open data analysis platform. PloS one.

[10] Szczepański, Amy F, Huang, J., Baer, T., & Mack, Y. C. (2013). Data analysis and visualization in high-performance computing. Computer, 46(5), 84-92.

[11] Stephan, S., Tobias, P., Stephan, P., & Pavel, T. (2013). Imglib2 for large scale image analysis and visualization. Frontiers in Neuroinformatics, 7(9), 559-569.

[12] Liu, H., Song, J., & Wang, G. (2021). A scientometric review of smart construction site in construction engineering and management: analysis and visualization. Sustainability, 13.

[13] Yaali, J., Vincent Grégoire, & Hurtut, T. (2022). Hftviz: visualization for the exploration of high frequency trading data:. Information Visualization, 21(2), 182-193.

[14] Machado, J. A. T., & Lopes, António M. (2013). Analysis and visualization of seismic data using mutual information. Entropy, 15(9), 3892-3909.

[15] Eren, A. M., Özcan C Esen, Quince, C., Vineis, J. H., & Delmont, T. O. (2015). Anvi'o: an advanced analysis and visualization platform for 'omics data. PeerJ, 3.

[16] Ben-Nun, T., & Hoefler, T. (2018). Demystifying parallel and distributed deep learning: an in-depth concurrency analysis. ACM Computing Surveys, 52(4).

[17] B, Y. L., B, Z. Z. A., A, X. L., A, L. W., & A, X. X. (2021). Ore image classification based on small deep learning model: evaluation and optimization of model depth, model structure and data size. Minerals Engineering.

About the Author

Ruixiang Guo, female, Han nationality, Zhangzhou Fujian, master's degree, the laboratory master, mainly engaged in the research of artificial intelligence, Information construction and Management Office, Min-nan Normal University.